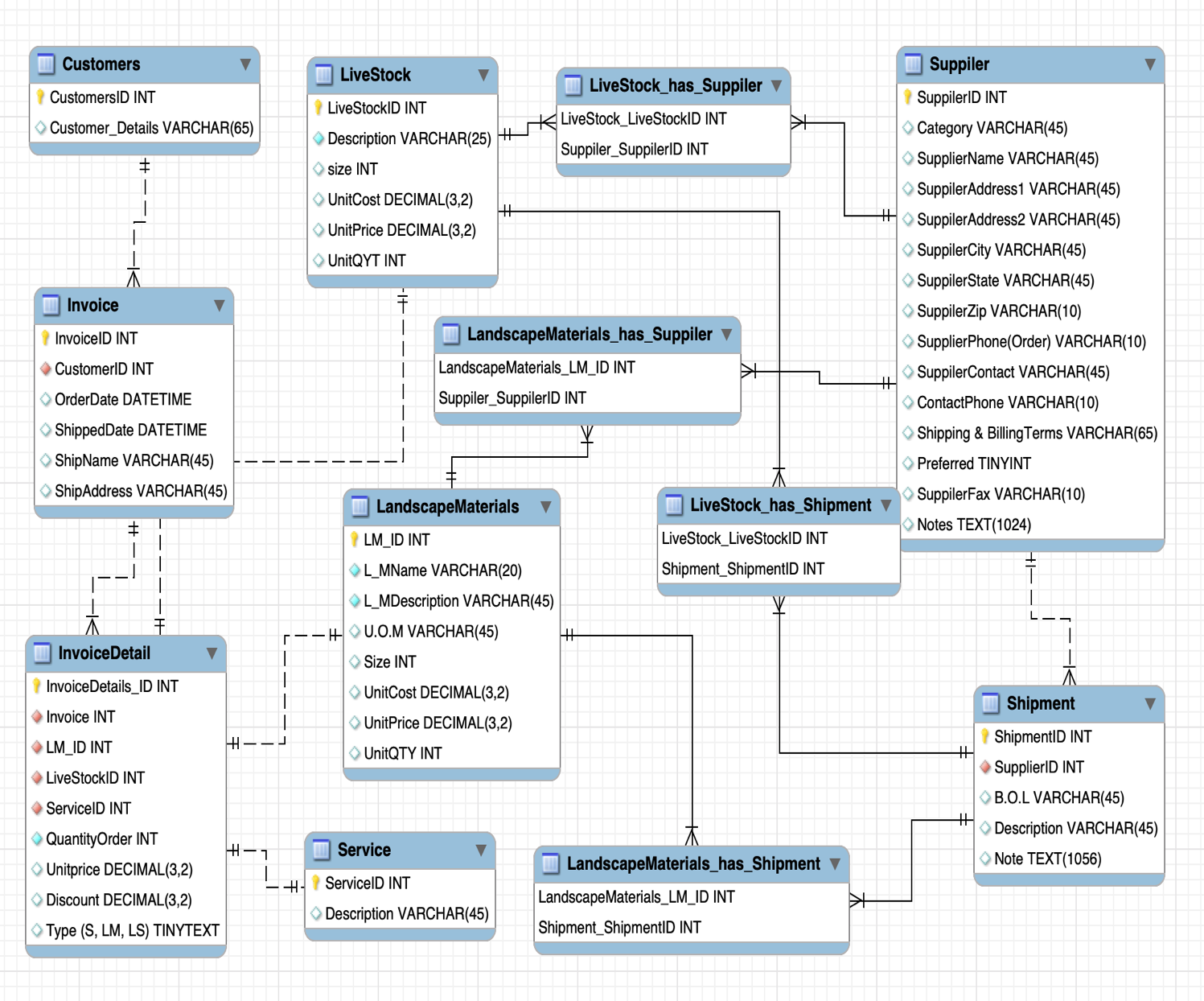
Name: Chakrya Ros

Homework 4 : Case Study

**Data Modal of Eden Landscape**



**Physical Data Model**

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema Eden\_landscape

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema Eden\_landscape

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `Eden\_landscape` DEFAULT CHARACTER SET utf8 ;

USE `Eden\_landscape` ;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`Suppiler`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`Suppiler` (

`SuppilerID` INT NOT NULL AUTO\_INCREMENT,

`Category` VARCHAR(45) NULL,

`SupplierName` VARCHAR(45) NULL,

`SuppilerAddress1` VARCHAR(45) NULL,

`SuppilerAddress2` VARCHAR(45) NULL,

`SuppilerCity` VARCHAR(45) NULL,

`SuppilerState` VARCHAR(45) NULL,

`SuppilerZip` VARCHAR(10) NULL,

`SupplierPhone(Order)` VARCHAR(10) NULL,

`SuppilerContact` VARCHAR(45) NULL,

`ContactPhone` VARCHAR(10) NULL,

`Shipping & BillingTerms` VARCHAR(65) NULL,

`Preferred` TINYINT NULL,

`SuppilerFax` VARCHAR(10) NULL,

`Notes` TEXT(1024) NULL,

PRIMARY KEY (`SuppilerID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`LandscapeMaterials`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`LandscapeMaterials` (

`LM\_ID` INT NOT NULL AUTO\_INCREMENT,

`L\_MName` VARCHAR(20) NOT NULL,

`L\_MDescription` VARCHAR(45) NOT NULL,

`U.O.M` VARCHAR(45) NULL,

`Size` INT NULL,

`UnitCost` DECIMAL(3,2) NULL,

`UnitPrice` DECIMAL(3,2) NULL,

`UnitQTY` INT NULL,

PRIMARY KEY (`LM\_ID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`LiveStock`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`LiveStock` (

`LiveStockID` INT NOT NULL AUTO\_INCREMENT,

`Description` VARCHAR(25) NOT NULL,

`size` INT NULL,

`UnitCost` DECIMAL(3,2) NULL,

`UnitPrice` DECIMAL(3,2) NULL,

`UnitQYT` INT NULL,

PRIMARY KEY (`LiveStockID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`Customers`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`Customers` (

`CustomersID` INT NOT NULL AUTO\_INCREMENT,

`Customer\_Details` VARCHAR(65) NULL,

PRIMARY KEY (`CustomersID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`Invoice`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`Invoice` (

`InvoiceID` INT NOT NULL,

`CustomerID` INT NOT NULL,

`OrderDate` DATETIME NULL,

`ShippedDate` DATETIME NULL,

`ShipName` VARCHAR(45) NULL,

`ShipAddress` VARCHAR(45) NULL,

PRIMARY KEY (`InvoiceID`),

INDEX `customerID\_idx` (`CustomerID` ASC) VISIBLE,

CONSTRAINT `customerID`

FOREIGN KEY (`CustomerID`)

REFERENCES `Eden\_landscape`.`Customers` (`CustomersID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`Service`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`Service` (

`ServiceID` INT NOT NULL AUTO\_INCREMENT,

`Description` VARCHAR(45) NULL,

PRIMARY KEY (`ServiceID`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`InvoiceDetail`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`InvoiceDetail` (

`InvoiceDetails\_ID` INT NOT NULL,

`Invoice` INT NOT NULL,

`LM\_ID` INT NOT NULL,

`LiveStockID` INT NOT NULL,

`ServiceID` INT NOT NULL,

`QuantityOrder` INT NOT NULL,

`Unitprice` DECIMAL(3,2) NULL,

`Discount` DECIMAL(3,2) NULL,

`Type (S, LM, LS)` TINYTEXT NULL,

PRIMARY KEY (`InvoiceDetails\_ID`),

INDEX `InvoiceID\_idx` (`Invoice` ASC) VISIBLE,

INDEX `LiveStockID\_idx` (`LiveStockID` ASC) VISIBLE,

INDEX `LM\_ID\_idx` (`LM\_ID` ASC) VISIBLE,

INDEX `ServiceID\_idx` (`ServiceID` ASC) VISIBLE,

CONSTRAINT `InvoiceID`

FOREIGN KEY (`Invoice`)

REFERENCES `Eden\_landscape`.`Invoice` (`InvoiceID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `LiveStockID`

FOREIGN KEY (`LiveStockID`)

REFERENCES `Eden\_landscape`.`LiveStock` (`LiveStockID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `LM\_ID`

FOREIGN KEY (`LM\_ID`)

REFERENCES `Eden\_landscape`.`LandscapeMaterials` (`LM\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `ServiceID`

FOREIGN KEY (`ServiceID`)

REFERENCES `Eden\_landscape`.`Service` (`ServiceID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`Shipment`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`Shipment` (

`ShipmentID` INT UNSIGNED NOT NULL AUTO\_INCREMENT,

`SupplierID` INT NOT NULL,

`B.O.L` VARCHAR(45) NULL,

`Description` VARCHAR(45) NULL,

`Note` TEXT(1056) NULL,

PRIMARY KEY (`ShipmentID`),

INDEX `SupplierID\_idx` (`SupplierID` ASC) VISIBLE,

CONSTRAINT `SupplierID`

FOREIGN KEY (`SupplierID`)

REFERENCES `Eden\_landscape`.`Suppiler` (`SuppilerID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`LiveStock\_has\_Suppiler`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`LiveStock\_has\_Suppiler` (

`LiveStock\_LiveStockID` INT NOT NULL,

`Suppiler\_SuppilerID` INT NOT NULL,

PRIMARY KEY (`LiveStock\_LiveStockID`, `Suppiler\_SuppilerID`),

INDEX `fk\_LiveStock\_has\_Suppiler\_Suppiler1\_idx` (`Suppiler\_SuppilerID` ASC) VISIBLE,

INDEX `fk\_LiveStock\_has\_Suppiler\_LiveStock1\_idx` (`LiveStock\_LiveStockID` ASC) VISIBLE,

CONSTRAINT `fk\_LiveStock\_has\_Suppiler\_LiveStock1`

FOREIGN KEY (`LiveStock\_LiveStockID`)

REFERENCES `Eden\_landscape`.`LiveStock` (`LiveStockID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_LiveStock\_has\_Suppiler\_Suppiler1`

FOREIGN KEY (`Suppiler\_SuppilerID`)

REFERENCES `Eden\_landscape`.`Suppiler` (`SuppilerID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`LiveStock\_has\_Shipment`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`LiveStock\_has\_Shipment` (

`LiveStock\_LiveStockID` INT NOT NULL,

`Shipment\_ShipmentID` INT UNSIGNED NOT NULL,

PRIMARY KEY (`LiveStock\_LiveStockID`, `Shipment\_ShipmentID`),

INDEX `fk\_LiveStock\_has\_Shipment\_Shipment1\_idx` (`Shipment\_ShipmentID` ASC) VISIBLE,

INDEX `fk\_LiveStock\_has\_Shipment\_LiveStock1\_idx` (`LiveStock\_LiveStockID` ASC) VISIBLE,

CONSTRAINT `fk\_LiveStock\_has\_Shipment\_LiveStock1`

FOREIGN KEY (`LiveStock\_LiveStockID`)

REFERENCES `Eden\_landscape`.`LiveStock` (`LiveStockID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_LiveStock\_has\_Shipment\_Shipment1`

FOREIGN KEY (`Shipment\_ShipmentID`)

REFERENCES `Eden\_landscape`.`Shipment` (`ShipmentID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`LandscapeMaterials\_has\_Suppiler`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`LandscapeMaterials\_has\_Suppiler` (

`LandscapeMaterials\_LM\_ID` INT NOT NULL,

`Suppiler\_SuppilerID` INT NOT NULL,

PRIMARY KEY (`LandscapeMaterials\_LM\_ID`, `Suppiler\_SuppilerID`),

INDEX `fk\_LandscapeMaterials\_has\_Suppiler\_Suppiler1\_idx` (`Suppiler\_SuppilerID` ASC) VISIBLE,

INDEX `fk\_LandscapeMaterials\_has\_Suppiler\_LandscapeMaterials1\_idx` (`LandscapeMaterials\_LM\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_LandscapeMaterials\_has\_Suppiler\_LandscapeMaterials1`

FOREIGN KEY (`LandscapeMaterials\_LM\_ID`)

REFERENCES `Eden\_landscape`.`LandscapeMaterials` (`LM\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_LandscapeMaterials\_has\_Suppiler\_Suppiler1`

FOREIGN KEY (`Suppiler\_SuppilerID`)

REFERENCES `Eden\_landscape`.`Suppiler` (`SuppilerID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Eden\_landscape`.`LandscapeMaterials\_has\_Shipment`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Eden\_landscape`.`LandscapeMaterials\_has\_Shipment` (

`LandscapeMaterials\_LM\_ID` INT NOT NULL,

`Shipment\_ShipmentID` INT UNSIGNED NOT NULL,

PRIMARY KEY (`LandscapeMaterials\_LM\_ID`, `Shipment\_ShipmentID`),

INDEX `fk\_LandscapeMaterials\_has\_Shipment\_Shipment1\_idx` (`Shipment\_ShipmentID` ASC) VISIBLE,

INDEX `fk\_LandscapeMaterials\_has\_Shipment\_LandscapeMaterials1\_idx` (`LandscapeMaterials\_LM\_ID` ASC) VISIBLE,

CONSTRAINT `fk\_LandscapeMaterials\_has\_Shipment\_LandscapeMaterials1`

FOREIGN KEY (`LandscapeMaterials\_LM\_ID`)

REFERENCES `Eden\_landscape`.`LandscapeMaterials` (`LM\_ID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_LandscapeMaterials\_has\_Shipment\_Shipment1`

FOREIGN KEY (`Shipment\_ShipmentID`)

REFERENCES `Eden\_landscape`.`Shipment` (`ShipmentID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**Thought and Process**

I assume that LiveStock, LandscapeMaterial entities have several supplier and shipment so I create the associate entities for those. LandscapeMaterial\_has\_supplier, LandscapeMaterial\_has\_shipment, LiveStock\_has\_supplier and LiveStockl\_has\_shipment are associate entities from many to one.